Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently amended) ~~In a computer system executing a software stream that removes a block of heap memory from a heap pile for use by a hardware device, a~~A method comprising:

performing, by a software stream, heap memory operations on a first end of a linked list of free heap memory of a heap pile; and concurrently

returning, by a hardware device, ~~the~~ a return block of heap memory to the heap pile at a second end of the linked list of free heap memory. ~~by the hardware device.~~

2.   (Cancelled).

3.   (Currently amended) The method as defined in claim [[2]] 1 wherein ~~placing~~ returning the return block of heap memory at the second end of the linked list ~~by the hardware device~~ further comprises:

writing a null to a next block field of the return block of heap memory;

writing a block number of the return block of heap memory to a next block field of a last block of heap memory in the linked list;

changing the contents of a bottom register to point to the return block of heap memory; and thereby

making the return block of heap memory a last entry in the linked list.

4.   (Currently amended) The method as defined in claim [[3]] 1 wherein performing heap memory operations further ~~comprising~~ comprises returning, by the software stream, a second block of heap memory by placing the second block of heap memory at ~~a beginning~~ the first end of the linked list.

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

5.   (Currently amended) The method as defined in claim 4 wherein ~~placing~~ returning the second block of heap memory at the ~~beginning~~ first end of the linked list by the software stream further comprises:

> determining a block number of a primary block of heap memory resident at the ~~beginning~~ first end of the linked list;
> writing the block number of the primary block of heap memory to a next block field of the second block; and
> writing atomically a block number of the second block to a top register.

6.   (Currently amended) The method as defined in claim 5 wherein determining a block number of a primary block of heap memory resident at the ~~beginning~~ first end of the linked list further comprises reading the top register prior to ~~the step of~~ writing the block number of the second block.

7.   (Currently amended) The method as defined in claim [[2]] 1 wherein performing heap memory operations further ~~comprising~~ comprises removing, by the software stream, heap memory from the first end of the linked list ~~heap management system~~.

8.   (Currently amended) The method as defined in claim 7 wherein removing heap memory from the linked list heap management system further comprises taking a primary block of heap memory resident at a ~~beginning~~ the first end of the of the linked list.

9.   (Currently amended) The method as defined in claim 8 wherein taking a primary block of heap memory ~~resident at a beginning of the~~ of the linked list further comprises:

> determining a block number of the primary block;
> reading a next block field of the primary block of memory; and
> removing the primary block if the next block field of the primary block does not indicate a null.

120988.01/1662.49800   Page 3 of 14

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

10.   (Original) The method as defined in claim 9 wherein determining a block number of the primary block further comprises reading a top register, wherein the top register identifies the beginning of the linked list.

11.   (Original) The method as defined in claim 9 wherein removing the primary block if the next block field of the primary block does not indicate a null further comprises writing a block number of the next block field of the primary block to the top register.

12.   (Currently amended) A method of managing a heap memory comprising:
> maintaining unused blocks of heap memory as a linked list, and wherein the unused blocks of the linked list comprise a first block at a beginning of the linked list, a second block pointed to the first block, and a third block at an end of the linked list;
> removing, by a software stream, the first block from the linked list, thus making the second block the beginning of the linked list; and
> returning, by a hardware device, a return block to the linked list by placing the return block at an the end of the linked list.

13.   (Original) The method of managing a heap memory as defined in claim 12 wherein returning, by a hardware device, a return block to the linked list by placing the return block at an end of the linked list further comprises:
> writing a null to a next block field of the return block;
> reading a bottom register, the bottom register identifying the third block;
> writing a block number of the return block to a next state field of the third block; and
> writing the block number of the return block to the bottom register.

14.   (Original) The method of managing a heap memory as defined in claim 12 wherein removing, by a software stream, the first block from the linked list further comprises:

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

reading a top register, the top register identifying the first block;

reading a next block field of the first block, the next block field of the first

block identifying the second block; and

writing a block number of the second block to the top register.

15.   (Original) The method of managing a heap memory as defined in claim 14
wherein writing a block number of the second block to the top register further
comprises atomically writing the block number of the second block to the top
register.

16.   (Original) The method of managing a heap memory as defined in claim 12
further comprising returning, by a software stream, a fourth block to the linked list
by placing the fourth block at the beginning of the linked list, thus making the
fourth block the beginning of the linked list.

17.   (Original) The method of managing a heap of memory as defined in claim
16 wherein returning a fourth block to the linked list by placing the fourth block at
the beginning of the linked list further comprises:

reading a top register, the top register identifying the beginning of the

linked list;

writing a block number of the block identified by the top register to a next

state field of the fourth block; and

writing a block number of the fourth block to the top register.

18.   (Original) The method of managing a heap memory as defined in claim 17
wherein writing a block number of the fourth block to the top register further
comprises atomically writing the block number of the fourth block to the top
register.

19.   (Original) A method of managing a heap memory in a computer system,
the method comprising:

120988.01/1662.49800   Page 5 of 14

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

            allowing a software thread to add and remove blocks of heap memory
               from a linked list of free blocks of heap memory in a last-in/first-out
               (LIFO) fashion at a first end of the linked list; and
            allowing a hardware device to add blocks of heap memory to the linked list
               of free blocks of heap memory at a second end of the linked list.

20.   (Original) The method of managing a heap memory in a computer system as defined in claim 19 wherein allowing a software thread to remove blocks of heap memory in LIFO fashion further comprises:

            determining, by the software thread, a block number of a block of heap
               memory at the first end of the linked list; and
            removing the block of heap memory at the first end of the linked list.

21.   (Original) The method of managing a heap memory in a computer system as defined in claim 20 determining a block number of a block of heap memory at the first end of the linked list further comprises reading a beginning register that stores a block number of a block of heap memory at the first end of the linked list.

22.   (Original) The method of managing a heap memory in a computer system as defined in claim 21 wherein removing the block of heap memory at the first end of the linked list further comprises:

            reading a next block field of the block of heap memory at the first end of
               the linked list to identify a block number of a next block in the linked
               list; and
            writing the block number of the next block in the linked list to the beginning
               register.

23.   (Original) The method of managing a heap memory in a computer system as defined in claim 20 wherein allowing a software thread to add blocks of heap memory in LIFO fashion further comprises:

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

> determining, by the software thread, a block number of a block of heap
>> memory at the first end of the linked list;
>
> writing the block number of the block of heap memory at the first end of the
>> linked list to a next block field of a return block of heap memory;
>> and
>
> making the return block of heap memory the first end of the linked list.

24.   (Original) The method of managing a heap memory in a computer system
as defined in claim 23 wherein determining a block number of a block of heap
memory at first end of the linked list further comprises reading a beginning
register that stores a block number of a block of heap memory at the first end of
the linked list.

25.   (Original) The method of managing a heap memory in a computer system
as defined in claim 24 wherein making the return block of heap memory the first
end of the linked list further comprises writing a block number the return block of
heap memory to the beginning register.

26.   (Original) The method of managing a heap memory in a computer system
as defined in claim 20 wherein allowing a hardware device to add blocks of heap
memory to the linked list of free blocks of heap memory at a second end of the
linked list further comprises:

> determining, by the hardware device, a block number of a block of heap
>> memory the second end of the linked list;
>
> writing, by the hardware device, a block number of a return block of heap
>> memory to a next block field of the block of heap memory at the
>> second end of the linked list; and
>
> making the return block of heap memory the second end of the linked list.

27.   (Original) The method of managing a heap memory in a computer system
as defined in claim 26 wherein determining a block number of a block of heap

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

memory at the second end of the linked list further comprises reading an end
register that stores a block number of the block of heap memory at the second
end of the linked list.

28.   (Original) The method of managing a heap memory in a computer system
as defined in claim 27 wherein making the return block of heap memory the
second end of the linked list further comprises writing a block number the return
block of heap memory to the end register.

29.   (Currently amended) A computer system comprising:
        a microprocessor executing a software stream;
        a main memory array, a portion of the main memory array allocated to be
            a heap memory, and wherein unused portions of the heap memory
            are part of a heap pile[[;]], the heap pile further comprising
                a plurality of blocks;
                each block having a next block field; and
                wherein the heap pile is maintained as a linked list, each
                    block's next block field pointing to a next block in the
                    list;
        a first bridge logic device coupling the microprocessor to the main memory
            array;
        a hardware device coupled to the heap memory through the first bridge
            logic device;
        a graphics card coupled to the first bridge logic device, and also coupled to
            a video display device;
        wherein the software stream executed on the microprocessor removes
            blocks of heap memory from a beginning of the heap pile for use by
            the hardware device; and simultaneously
        wherein the hardware device returns blocks of heap memory to an end of
            the heap pile without intervention from the software stream
            executed by the microprocessor.

120988.01/1662.49800   Page 8 of 14

Appl. No.: 10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

30. (Currently amended) The computer system as defined in claim 29 wherein
the heap memory further comprises:
 a plurality of blocks, each block having each have the same number of
 bytes;
 each block having a next block field; and
 wherein the heap pile is maintained as a linked list, each block's next block
 filed pointing to a next block in the list.

31. (Cancelled).

32. (Currently amended) The computer system as defined in claim 31 29
further comprising the software stream returns blocks to the heap pile at the top
end beginning of the heap pile simultaneously as the hardware device returns
blocks of heap memory to the end of the heap pile.

33. (Original) The computer system as defined in claim 29 wherein the
hardware device is the graphics card.

34. (Original) The computer system as defined in claim 29 wherein the
hardware device is a network interface card.

35. (Original) The computer system as defined in claim 29 wherein the
hardware device is an audio card.

36. (Original) The computer system as defined in claim 29 wherein the
hardware device is a mass storage device.

37. (Original) The computer system as defined in claim 36 wherein the mass
storage device is a hard drive.

120988.01/1662.49800   Page 9 of 14

Appl. No.:   10/037,806
Amdt. dated March 11, 2004
Reply to Office Action of January 29, 2004

38.   (Original) The computer system as defined in claim 37 wherein the mass storage device is compact disk storage device.